Name: C. Shyam Ganesh
RegNo: 192324292
Course code: CSA 0666
Course Name: DAA
Date: 08.06.2024
Assignment NO: 2

1) If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$, then $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$. Prove the assertions.

We know, $t_1(n) \le c_1 \cdot g_1(n)$ for all $n \ge n_1$
and $t_2(n) \le c_2 \cdot g_2(n)$ for all $n \ge n_2$

Let $n_0 = \max\{n_1, n_2\}$ for all $n \ge n_0$

Consider the sum, $t_1(n) + t_2(n)$ for all $n \ge n_0$

we have, $t_1(n) + t_2(n) \le c_1 \cdot g_1(n) + c_2 \cdot g_2(n)$

we assume $g_1(n) \ge g_2(n)$ then $\max\{g_1(n), g_2(n)\} = g_1(n)$

$\Rightarrow c_1 \cdot g_1(n) + c_2 \cdot g_2(n) \le c_1 \cdot g_1(n) + c_2 \cdot g_2(n) = c_1 [\cancel{g_1(n) + g_2(n)}]$

$\Rightarrow c_1 \cdot g_1(n) + c_2 \cdot g_2(n) \le g_1(n)[c_1 + c_2]$

$\because g_1(n) = \{\max\{g_1(n), g_2(n)\}\} = g_1(n)$

$\Rightarrow t_1(n) + t_2(n) \le (c_1 + c_2) \cdot \max\{g_1(n), g_2(n)\}$ for all $n \ge n_0$

$\therefore \boxed{t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})}$

Hence proved

2) Find Time complexity of below recurrence relation.

3) $T(n) = \begin{cases} 2T(n/2) + 1 & \text{if } n > 1 \\ 1 & \text{otherwise} \end{cases}$

$a = 2, \; b = 2, \; k = 0, \; p = 0$

$\log_b a = \log_2 2 = 1, \; k = 0$

Here $k < \log_b a$

$\therefore T(n) = \Theta[n^k \log_n \cancel{(\ )}_b^a]$

$f(n) = 1, \; n^{\log_b a} = n^1 = n$

$f(n) = 1 = O(n^c)$ where $c = 0$ and $0 < 1$

$\Rightarrow T(n) = \Theta(n^{\log_b a})$

$\Rightarrow \boxed{T(n) = \Theta(n)}$

4) $T(n) = \begin{cases} 2T(n-1) & , n>0 \\ 1 & , \text{otherwise} \end{cases}$

Sol: $T(n) = 2T(n-1)$

put $n = n-1$, then

$T(n) = 2 \cdot 2T(n-2) = 2^2 T(n-2)$

continuing the process, we get

$2^K T(n-K)$

If $K = n$, then

$T(n) = 2^n T(n-n) = 2^n T(0)$ $\quad \because T(0) = 1$

$\Rightarrow T(n) = 2^n \cdot 1 = 2^n$

$\therefore \boxed{T(n) = O(2^n)}$

---

5) Big O notation: Show that $f(n) = n^2+3n+5$ is $O(n^2)$

Let $f(n) \le c \cdot g(n)$ where $g(n) = n^2$

$\Rightarrow n^2+3n+5 \le c \cdot n^2$

$\Rightarrow 1 + \frac{3}{n} + \frac{5}{n^2} \le c$

for $n \ge 1$, $3/n \le 3$ and $\frac{5}{n^2} \le 5$

$\Rightarrow 1 + \frac{3}{n} + \frac{5}{n^2} \le 1+3+5 = 9$

So, we can choose $c = 9$

$\therefore$ for $n \ge 1$, $1 + \frac{3}{n} + \frac{5}{n^2} \le 9$

$\Rightarrow n^2+3n+5 \le 9n^2 = O(n^2)$ with $c = 9$, $n_0 = 1$

---

6) Big Omega notation: $g(n) = n^3+2n^2+4n$ is $\Omega(n^3)$

Let $n^3+2n^2+4n \ge c \cdot n^3$

then for $n \ge 1$, $2n^2 \le n^3$ and $4n \le n^3$

(i.e) $2n^2 \le n^3$ for $n \ge 2$

$4n \le n^3$ for $n \ge 4$

So, for $n \ge 4$

$2n^2+4n \le n^3+n^3 = 2n^3$

$\Rightarrow$ for $n \ge 4$, $g(n) = n^3+2n^2+4n \ge n^3 = \Omega(n^3)$

7) Big Theta Notation: prove: $h(n) = 4n^2+3n$ is $\Theta(n^2)$

Let $4n^2+3n \le c \cdot n^2$ for $n \ge n_0$

So, for $n \ge 1$, $3n \le 4n^2$

$\Rightarrow 4n^2+3n \le 4n^2+3n^2$ ($\because 3n \le 3n^2$ for $n \ge 1$)

$\Rightarrow 4n^2+3n \le 7n^2$

$\Rightarrow c = 7, n_0 = 1$

$\therefore h(n) \le 7n^2$ for all $n \ge 1$ $\in O(n^2) \rightarrow (1)$

Let $4n^2+3n \ge c \cdot n^2$ for $n \ge n_0$

we choose $c = 1$, $n_0 = 1$, then

$h(n) \ge 4n^2$ for all $n \ge 1$ $\in \Omega(n^2) \rightarrow (2)$

As $h(n)$ satisfies both Upper as well as lower bound.

Hence $\boxed{h(n) \in \Theta(n^2)}$

$\therefore$ Time complexity: $\boxed{\Theta(n^2)}$

8) Let $f(n) = n^3-2n^2+n$ and $g(n) = n^2$ show whether $f(n) = \Omega(g(n))$ is true or false. Justify.

Let $f(n) \ge -c \cdot n^2$

$\Rightarrow n^3-2n^2+n \ge -c \cdot n^2$

$\Rightarrow n^3-2n^2+n+cn^2 \ge 0$

$\Rightarrow n^3+(c-2)n^2+n \ge 0$

for $n \ge n_0$, $n^3+(c-2)n^2+n \approx n^3$

So, choosing $n_0 = 1$ and $c = 1$, we get

$n^3+(1-2)n^2+n \ge 0$

$\therefore f(n) \ge c \cdot g(n)$ is true $= \Omega(g(n))$ for all values of $n > 0$ and $n \ge n_0$

Hence proved.

9) Determine whether $h(n) = n\log n + n$ is in $\Theta(n\log n)$

Let, $h(n) \le c \cdot n\log n$.

$\Rightarrow n\log n + n \le c \cdot n\log n$ ~~for large~~

for large values of n, $\log n$ will dominate, so

$n\log n + n \le 2n\log n$ for $n \ge 1$

$\Rightarrow h(n) = n\log(n) + n \in O(n\log n)$

$n\log n + n \geq c \cdot n\log n$

Let n_0 assume, for $n \geq 1$, $1 + \frac{1}{\log n} \leq 2$

$\Rightarrow n\log n(1 + \frac{1}{\log n})$ $c = 1$, $n_0 = 2$, we get

$\Rightarrow n\log n(1 + \frac{1}{\log n}) \geq n\log n$ for all $n \geq 1$ and $n > n_0$

$h(n) \geq n\log n$ $\in \Omega(n\log n)$

$\Rightarrow h(n) = n\log n + n$

$\therefore$ $\boxed{\text{Time complexity} : \Theta(n\log n)}$

10) Solve: $T(n) = 4T(n/2) + n^2$, $T(1) = 1$

$a = 4$, $b = 2$, $k = 2$, $p = 1$

$\log_b a = \log_2 4 = \boxed{2}$

here $k = \log_b a$ and $p = 1$, so,

$T(n) = \Theta(n^k \log^{p+1} n)$ $T(n) = \Theta(n^k \log n)$

$T(n) = \Theta(n^k \log_n 1)$

$\Rightarrow$ $\boxed{T(n) = \Theta(n^2 \log n)}$

11) Given an array [4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 9, -1, 0, -8, 11, 9] find max and min product

Algorithm:

```
def p(a):
    if length(a) < 2:
        return -1
    max1=0, max2=0, min1=0, min2=0
    for i in a:
        if i > max1:
            max2 = max1
            max1 = i
        elif i > max2:
            max2 = i
        if i < min1:
            min2 = min1
            min1 = i
        elif i < min2:
            min2 = i
maxP = max(max1 * min2, min1 * min2)
minP = min(max1 * min1, max2 * min2)
```

12) Demonstrate Binary Search method to search key = 23 in the array a[] = { 2, 5, 8, 12, 16, 23, 38, 56, 72, 91 }

Demonstration:

```
a = [2,5,8,12,16,23,38,56,72,91]
key = 23
left = 0
right = len(a)-1
while left <= right:
      mid = left + (right-left) // 2
      if a[mid] == key:
            print("key", key, "is found at", mid)
      elif a[mid] < key:
            left = mid+1
      else:
          right = mid-1
else:
   print("Not found")
```

---

13) Apply merge sort and order the list of 8 elements. Set up a recurrence relation for no. of key comparisons made by mergesort.

```
def m(a):
    if len(a)>1:
        mid = len(a)//2
        l = a[:mid]
        r = a[mid:]
        m(l)
        m(r)
        i=j=0
        k=0
        while(i<len(l) and j<len(r)]:
            if l[i] < r[j]:
                a[k] = l[i]
                i+=1
            else:
                a[k] = r[j]
                j+=1
        k+=1
    while i<len(l):
        a[k] = l[i]
```

```
i+=1
k+=1
while j < len(r):
    a[k]=r[j]
    j+=1
    k+=1
a=[45,67,-12,5,22,30,50,20]
m(a)
print(a)
```

In merge sort as the no. of elements → n is divided into n/2 (divide) and then become n again (conquer) after finding solution

$\therefore \boxed{T(n) = 2T(n/2)+n}$

(14) Find the no. of times to perform swapping for selection sort. Also find Time complexity.

set S(12, 7, 5, -2, 18, 6, 13, 4)

Algorithm:

```
def s(a):
  swap=0
  n=length(a)
  for i in range(n):
      min=i
      for j in range(i+1,n):
          if a[j] < a[min]
              min=j
      if min !=i
          a[i], a[min] = a[min], a[i]
          swap += 1
  return swap

n=s(a)
print(a)
```

No. of times of swap : 4

Time complexity : $O(n^2)$

15) Find the
from the
Algorithm
a=[2,4,
key=
left=0
n=len
right=
if le

An

16)
[S
co

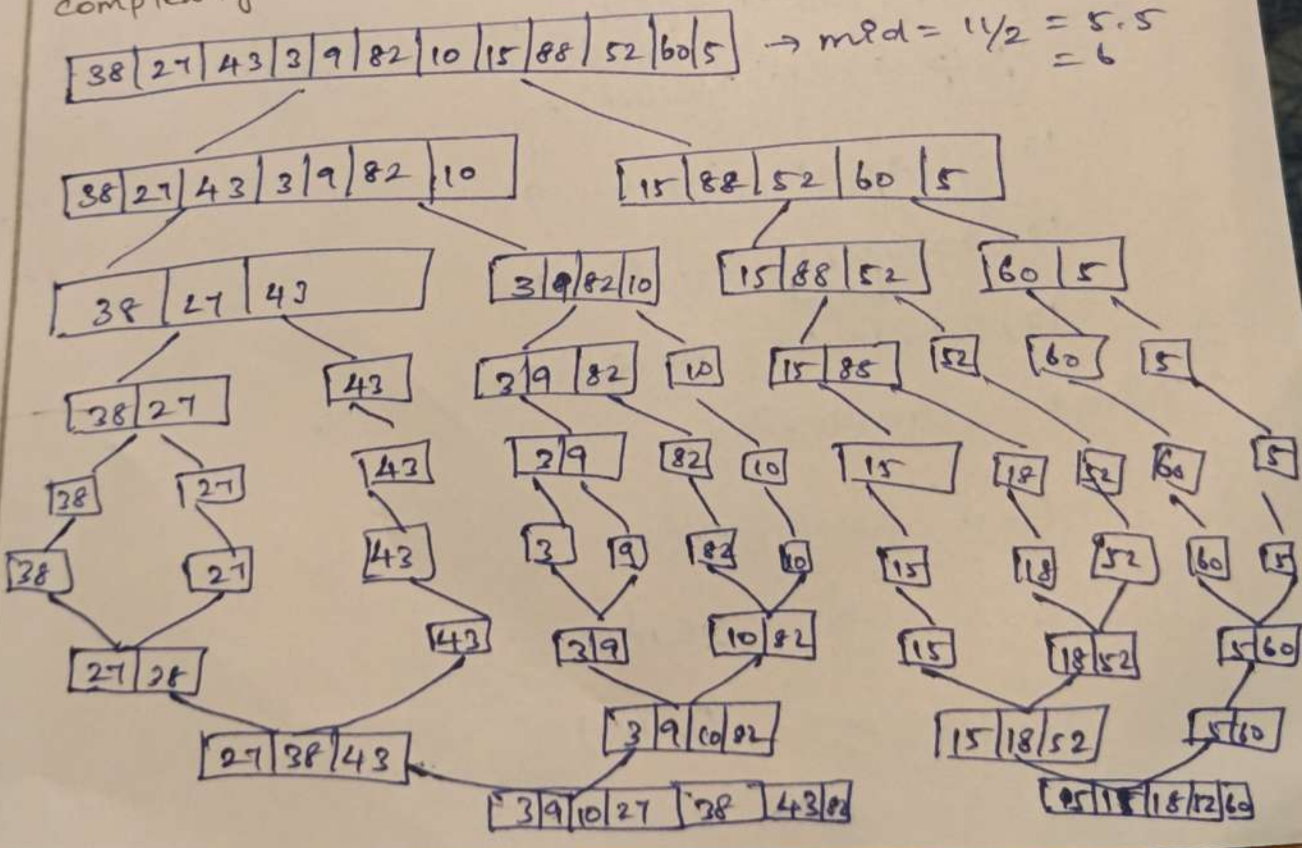
15) Find the index of value 10 using binary search from the following elements [2,4,6,8,10,12,14,16,18,20]

Algorithm :

```
a = [2,4,6,8,10,12,14,16,18,20]
key = 10
left = 0
n = len(a)
right = n-1
if left <= right :
    mid = left + (right - left)//2
    if a[mid] == key:
        print(key, "is found at", mid, "position")
    elif a[mid] < key:
        left = mid+1
    elif a[mid] > key:
        right = mid-1
    else:
        print("Not found")
```

Ans: 10 is found at 4 position

16) Sort the following using Merge sort divide and conquer [38,27,43,3,9,82,10,15,88,52,60,5] and analyse time complexity



For dividing , time complexity becomes O(log n)
for conquer , time complexity becomes O(n)

∴ Time complexity : O(n log n)

17) a = [64, 34, 25, 12, 22, 11, 90] sort using bubble sort . Also find time complexity.

Case I :

64, 34, 25, 12, 22, 11, 90

34, 64, 25, 12, 22, 11, 90

34, 25, 64, 12, 22, 11, 90

34, 25, 12, 64, 22, 11, 90

34, 25, 12, 22, 64, 11, 90

34, 25, 12, 22, 11, 64, 90

case II :

34, 25, 12, 22, 11, 64, 90

25, 34, 12, 22, 11, 64, 90

25, 12, 34, 22, 11, 64, 90

25, 12, 22, 34, 11, 64, 90

25, 12, 22, 11, 34, 64, 90

case III :

25, 12, 22, 11, 34, 64, 90

12, 25, 22, 11, 34, 64, 90

12, 22, 25, 11, 34, 64, 90

12, 22, 11, 25, 34, 64, 90

case IV :

12, 22, 11, 25, 34, 64, 90

12, 11, 22, 25, 34, 64, 90

case V :

12, 11, 22, 25, 34, 64, 90

$\boxed{11, 12, 22, 25, 34, 64, 90}$

↳ Ans:

Time complexity :

Best case : O(n)

Avg case : O(n²)

Worst case : O(n²)

18) a = [64, 25, 12, 22, 11] sort using selection sort. Also find time complexity.

Selection sort:

64	25	12	22	11
11	25	12	22	64
11	12	25	22	64
11	12	22	25	64

Time complexity:

Best case: $O(n^2)$

Avg case: $O(n^2)$

Worst case: $O(n^2)$

19) Sort the following elements using Insertion sort using Brute force Approach strategy [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] and analyse complexity.

Insertion sort:

38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5

27, 38, 43, 3, 9, 82, 10, 15, 88, 52, 60

27, 38, 3, 43, 9, 82, 10, 15, 88, 52, 60

27, 3, 38, 43, 9, 82, 10, 15, 88, 52, 60

3, 27, 38, 43, 9, 82, 10, 15, 88, 52, 60

3, 27, 38, 9, 43, 82, 10, 15, 88, 52, 60

3, 27, 9, 38, 43, 82, 10, 15, 88, 52, 60

3, 9, 27, 38, 43, 82, 10, 15, 88, 52, 60

3, 9, 27, 38, 43, 10, 82, 15, 88, 52, 60

3, 9, 27, 38, 10, 43, 82, 15, 88, 52, 60

3, 9, 27, 10, 38, 43, 82, 15, 88, 52, 60

3, 9, 10, 27, 38, 43, 82, 15, 88, 52, 60

3,9,10,27,38,43,15,82,88,52,60

3,9,10,27,38,15,43,82,88,52,60

3,9,10,27,15,38,43,82,88,52,60

3,9,10,15,27,38,43,82,88,52,60

3,9,10,15,27,38,43,82,52,88,60

3,9,10,15,27,38,43,52,82,88,60

3,9,10,15,27,38,43,52,82,60,88

3,9,10,15,27,38,43,52,60,82,88

Time complexity: O(n)

Space complexity: O(1)

---

20) Solve the following elements using Brute Force Approach strategy and analyse complexity.

a=[4,-2,5,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9]

Insertion sort:

4,-2,5,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-2,4,5,3,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-2,4,3,5,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-2,3,4,5,10,-5,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-2,3,4,5,-5,10,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-2,3,4,-5,5,10,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-2,3,-5,4,5,10,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-2,-5,3,4,5,10,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,3,4,5,10,2,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,3,4,5,2,10,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,3,4,2,5,10,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,3,2,4,5,10,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,3,4,5,10,8,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,3,4,5,8,10,

-5,-2,2,3,4,5,8,

-5,-2,2,3,4,5,

-5,-2,2,3,4,-3,

-5,-2,2,3,-3,

-5,-2,2,-3,3,

-5,-2,-3,2

-5,-3,-2,2,

-5,-3,-2,

-5,-3,-2,

-5,-3,-2,

-5,-3,

-5,-3,

-5,-3

-5,-3,

-5,-3

-5,-3

-5,-3

AM :

Time

Space

-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,3,4,5,8,10,-3,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,3,4,5,8,-3,10,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,3,4,5,-3,8,10,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,3,4,-3,5,8,10,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,3,-3,4,5,8,10,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,2,-3,3,4,5,8,10,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-2,-3,2,3,4,5,8,10,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,8,10,6,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,8,6,10,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,6,8,10,7,-4,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,6,8,7,10,-4,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,6,7,8,10,-4,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,6,7,8,-4,10,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,6,7,-4,8,10,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,6,-4,7,8,10,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,5,-4,6,7,8,10,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,4,-4,5,6,7,8,10,1,9,-1,0,-6,-8,11,-9

-5,-3,-2,2,3,-4,4,5,6,7,8,10,1,9,-1,0,-6,-8,11,-9 ...

Ans: -8,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11

Time complexity: O(n)

Space complexity: O(1)